

# RESTful Services

Funshine

# Teil 5: Wetter in Listenform

# Hinzufügen der Libraries

```
dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.3.1'
    testCompile 'junit:junit:4.12'
    compile 'com.android.support.constraint:constraint-layout:1.0.2'
    compile 'com.android.volley:volley:1.0.0'
    //compile 'com.google.android.gms:play-services:10.2.1'
    compile 'com.google.android.gms:play-services-maps:10.2.1'
    compile 'com.google.android.gms:play-services-location:10.2.1'
    compile 'com.android.support:recyclerview-v7:25.3.1'
    compile 'com.android.support:cardview-v7:25.3.1'
}
```

# RecyclerView

The image shows the Android Studio IDE interface for a weather application. The main window displays a preview of the app on a Nexus 4 device. The app's layout includes a dark blue header with the text "Funshine", a light blue background with a yellow sun icon, and weather information for "Portland, USA" on "Today, May 1". The current temperature is "78°" and the low is "64°". The weather description is "Clear".

The left sidebar contains the "Palette" and "Component Tree" panels. The "Palette" panel shows a list of Android widgets, with "RecyclerView" highlighted in blue. The "Component Tree" panel shows the hierarchy of the app's layout, with "RecyclerView" listed at the bottom. The "Component Tree" also shows several warning icons (yellow triangles with exclamation marks) next to the "ic\_weather\_icon\_mini", "ic\_weather\_logo", "ic\_weather\_icon", and "RecyclerView" components.

The "Component Tree" structure is as follows:

- ConstraintLayout
  - ic\_weather\_icon\_mini (ImageView) ⚠
  - ic\_weather\_logo (ImageView) ⚠
  - tv\_weather\_date (TextView) - "Today, May 1" ⚠
  - tv\_current\_temp (TextView) - "78°" ⚠
  - tv\_low\_temp (TextView) - "64°" ⚠
  - ic\_weather\_icon (ImageView) ⚠
  - tv\_city\_country (TextView) - "Portland, USA" ⚠
  - tv\_weather\_description (TextView) - "Clear" ⚠
  - RecyclerView



```
<android.support.v7.widget.RecyclerView  
    android:id="@+id/content_weather_report"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginEnd="8dp"  
    android:layout_marginStart="8dp"  
    android:layout_marginTop="16dp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/tv_city_country"  
    android:layout_marginBottom="16dp" />
```

```
</android.support.constraint.ConstraintLayout>
```

Android

- app
  - manifests
    - AndroidManifest.xml
  - java
    - at.htl.funshine
      - model
        - DailyWeatherRepo
        - WeatherActivity
      - at.htl.funshine (androidTest)
      - at.htl.funshine (test)
  - res
    - drawable
    - layout
      - activity\_weather.xml
    - mipmap
    - values
  - Gradle Scripts
    - build.gradle (Project: Funshine)
    - build.gradle (Module: app)
    - gradle-wrapper.properties (Gr)
    - proguard-rules.pro (ProGuard)
    - gradle.properties (Project Prop)
    - settings.gradle (Project Setting)
    - local.properties (SDK Location)

```
WeatherActivity weatherReportViewHolder updateUI()
193 @Override
194 public void onConnectionFailed(@NonNull ConnectionResult connectionResult) {
195
196 }
197
198 @Override
232 }
233 }
234 }
235
236 public class weatherReportViewHolder extends RecyclerView.ViewHolder {
237
238     public weatherReportViewHolder(View itemView) {
239         super(itemView);
240     }
241
242     public void updateUI(DailyWeatherReport report) {
243
244 }
```

### New Resource File

File name:

Root element:

Source set:

Directory name:

Available qualifiers:

- Country Code
- Network Code
- Locale
- Layout Direction
- Smallest Screen Width
- Screen Width
- Screen Height
- Size
- Ratio
- Orientation
- UI Mode
- Night Mode
- Density

Chosen qualifiers:

Nothing to show

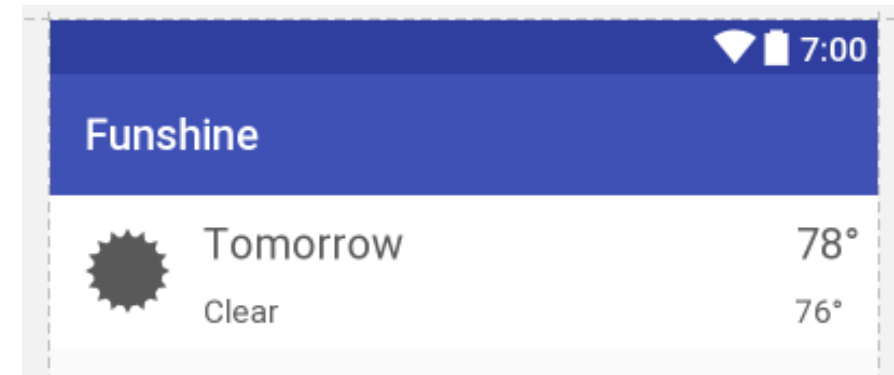
Buttons: >> <<

Buttons: Cancel OK

# card\_weather.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<android.support.v7.widget.CardView  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:card_view="http://schemas.android.com/apk/res-auto"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:background="#ECECEC">  
  
</android.support.v7.widget.CardView>
```

# card\_weather.xml



```
<android.support.constraint.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#FFF">

    <ImageView
        android:id="@+id/ic_list_weather_icon"
        android:layout_width="55dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        card_view:layout_constraintBottom_toBottomOf="parent"
        card_view:layout_constraintStart_toStartOf="parent"
        card_view:layout_constraintTop_toTopOf="parent"
        card_view:srcCompat="@drawable/sunny_mini" />

    <TextView
        android:id="@+id/tv_list_weather_date"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        android:text="Tomorrow"
        android:textColor="#595959"
        android:textSize="20sp"
        card_view:layout_constraintBottom_toTopOf="@+id/
tv_list_weather_description"
        card_view:layout_constraintStart_toEndOf="@+id/
ic_list_weather_icon"
        card_view:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
    android:id="@+id/tv_list_weather_description"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginStart="8dp"
    android:text="Clear"
    android:textColor="#595959"
    android:textSize="15sp"
    card_view:layout_constraintBottom_toBottomOf="parent"
    card_view:layout_constraintStart_toEndOf="@+id/ic_list_weather_icon" />

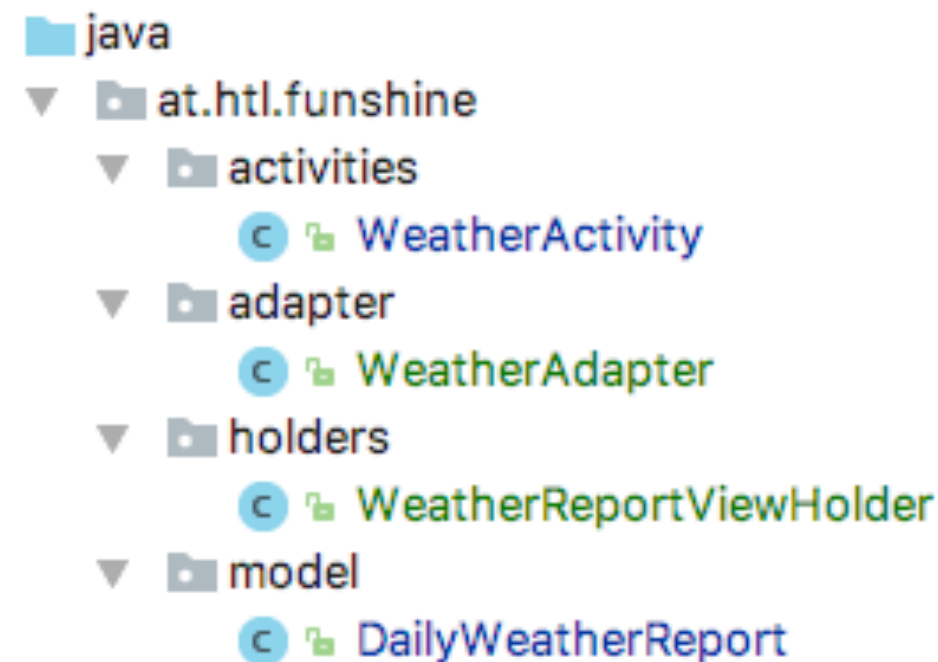
    <TextView
        android:id="@+id/tv_list_weather_temp_high"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:text="78°"
        android:textColor="#595959"
        android:textSize="20sp"
        card_view:layout_constraintBaseline_toBaselineOf="@+id/
tv_list_weather_date"
        card_view:layout_constraintEnd_toEndOf="parent"
        card_view:layout_constraintHorizontal_bias="1.0"
        card_view:layout_constraintStart_toEndOf="@+id/tv_list_weather_date" />

    <TextView
        android:id="@+id/tv_list_weather_temp_low"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="14dp"
        android:layout_marginRight="16dp"
        android:text="76°"
        android:textColor="#595959"
        android:textSize="15sp"
        card_view:layout_constraintBaseline_toBaselineOf="@+id/
tv_list_weather_description"
        card_view:layout_constraintEnd_toEndOf="parent" />
</android.support.constraint.ConstraintLayout>
</android.support.v7.widget.CardView>
```



# Struktur

- Es werden nun noch zwei Klassen angelegt:
  - WeatherAdapter
  - WeatherReportViewHolder



# WeatherAdapter.java

```
public class WeatherAdapter extends RecyclerView.Adapter<WeatherReportViewHolder> {  
  
    private List<DailyWeatherReport> mDailyWeatherReports;  
  
    public WeatherAdapter(List<DailyWeatherReport> dailyWeatherReports) {  
        this.mDailyWeatherReports = dailyWeatherReports;  
    }  
  
    @Override  
    public void onBindViewHolder(WeatherReportViewHolder holder, int position) {  
        DailyWeatherReport report = mDailyWeatherReports.get(position);  
        holder.updateUI(report);  
    }  
  
    @Override  
    public int getItemCount() {  
        return mDailyWeatherReports.size();  
    }  
  
    @Override  
    public WeatherReportViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {  
        View card = LayoutInflater  
            .from(parent.getContext())  
            .inflate(R.layout.card_weather, parent, false);  
        return new WeatherReportViewHolder(card);  
    }  
}
```

# WeatherReportViewHolder.java

```
public class WeatherReportViewHolder extends RecyclerView.ViewHolder {

    private ImageView listWeatherIcon;
    private TextView listWeatherDate;
    private TextView listWeatherDescription;
    private TextView listTempHigh;
    private TextView listTempLow;

    public WeatherReportViewHolder(View itemView) {
        super(itemView);

        listWeatherIcon = (ImageView) itemView.findViewById(R.id.ic_list_weather_icon);
        listWeatherDate = (TextView) itemView.findViewById(R.id.tv_list_weather_date);
        listWeatherDescription = (TextView) itemView.findViewById(R.id.tv_list_weather_description);
        listTempHigh = (TextView) itemView.findViewById(R.id.tv_list_weather_temp_high);
        listTempLow = (TextView) itemView.findViewById(R.id.tv_list_weather_temp_low);
    }

    public void updateUI(DailyWeatherReport report) {

        listWeatherDate.setText(report.getFormattedDate());
        listWeatherDescription.setText(report.getWeather());
        listTempHigh.setText(Integer.toString(report.getMaxTemp()));
        listTempLow.setText(Integer.toString(report.getMinTemp()));

        switch (report.getWeather()) {
            case DailyWeatherReport.WEATHER_TYPE_CLOUDS:
                listWeatherIcon.setImageDrawable(listWeatherIcon.getResources().getDrawable(R.drawable.cloudy_mini));
                break;
            case DailyWeatherReport.WEATHER_TYPE_RAIN:
                listWeatherIcon.setImageDrawable(listWeatherIcon.getResources().getDrawable(R.drawable.rainy_mini));
                break;
            default:
                listWeatherIcon.setImageDrawable(listWeatherIcon.getResources().getDrawable(R.drawable.sunny_mini));
        }
    }
}
```

# WeatherActivity.java

```
public class WeatherActivity extends AppCompatActivity
    implements GoogleApiClient.OnConnectionFailedListener,
    GoogleApiClient.ConnectionCallbacks,
    LocationListener {

    final String LOG_TAG = AppCompatActivity.class.getSimpleName();
    final String URL_BASE = "http://api.openweathermap.org/data/2.5/forecast";
    final String URL_COORD = "?lat="; //"?lat=48.2686066&lon=14.2493933";
    final String URL_UNITS = "&units=metric";
    final String URL_API_KEY = "&APPID=5b59acdd3bf5119d2fd7f1f958ae01ec";
    //final String URL_API_KEY = "&APPID=VerwendeDeinenAPI-Key";

    private GoogleApiClient mGoogleApiClient;
    private final int PERMISSION_LOCATION = 111;
    private List<DailyWeatherReport> weatherReportList = new ArrayList<>();

    private ImageView weatherIcon;
    private ImageView weatherIconMini;
    private TextView weatherDate;
    private TextView currentTemp;
    private TextView lowTemp;
    private TextView cityCountry;
    private TextView weatherDescription;

    WeatherAdapter mAdapter;
```

# WeatherActivity.java

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_weather);
```

```
    weatherIcon = (ImageView) findViewById(R.id.ic_list_weather_icon);  
    weatherIconMini = (ImageView) findViewById(R.id.ic_weather_icon_mini);  
    weatherDate = (TextView) findViewById(R.id.tv_weather_date);  
    currentTemp = (TextView) findViewById(R.id.tv_current_temp);  
    lowTemp = (TextView) findViewById(R.id.tv_low_temp);  
    cityCountry = (TextView) findViewById(R.id.tv_city_country);  
    weatherDescription = (TextView) findViewById(R.id.tv_list_weather_description);
```

```
    RecyclerView recyclerView = (RecyclerView) findViewById(R.id.content_weather_report);
```

```
    mAdapter = new WeatherAdapter(weatherReportList);  
    recyclerView.setLayoutManager(new LinearLayoutManager(this));  
    recyclerView.setAdapter(mAdapter);
```

```
    mGoogleApiClient = new GoogleApiClient.Builder(this)  
        .addApi(LocationServices.API)  
        .enableAutoManage(this, this)  
        .addConnectionCallbacks(this)  
        .addOnConnectionFailedListener(this)  
        .build();
```

```
}
```

# WeatherActivity.java

```
public void downloadWeatherData(Location location) {
    final String fullCoords = URL_COORD + location.getLatitude() + "&lon=" +
location.getLongitude();
    // final String fullCoords = "?lat=25.778&lon=-80.2018"; // Miami
    final String url = URL_BASE + fullCoords + URL_UNITS + URL_API_KEY;

    final JsonObjectRequest jsonRequest = new JsonObjectRequest(
        Request.Method.GET,
        url,
        null, // hier könnte man JSON-Daten im Body mitschicken
        new Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject response) {
                try {
                    ...
                } catch (JSONException e) {
                    Log.e(LOG_TAG, e.getLocalizedMessage());
                }

                updateUI();
                mAdapter.notifyDataSetChanged();
            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Log.v(LOG_TAG, "Err: " + error.getLocalizedMessage());
            }
        }
    });
};
```

try-Block: nächste Folie

# WeatherActivity.java

```
JSONObject city = response.getJSONObject("city");
String cityName = city.getString("name");
String country = city.getString("country");

JSONArray list = response.getJSONArray("list");
for (int i = 0; i < list.length(); i += 8) {
    JSONObject obj = list.getJSONObject(i);
    JSONObject main = obj.getJSONObject("main");
    Double currentTemp = main.getDouble("temp");
    Double maxTemp = main.getDouble("temp_max");
    Double minTemp = main.getDouble("temp_min");

    JSONArray weatherArr = obj.getJSONArray("weather");
    JSONObject weather = weatherArr.getJSONObject(0);
    String weatherType = weather.getString("main");

    String rawDate = obj.getString("dt_txt");

    DailyWeatherReport report = new DailyWeatherReport(
        cityName, country, currentTemp.intValue(),
        maxTemp.intValue(), minTemp.intValue(), weatherType, rawDate
    );
    Log.v(LOG_TAG, "Printing vom class: " + report.getWeather());
    weatherReportList.add(report);
}

Log.v(LOG_TAG, "Name: " + cityName + " - Country: " + country);
```

try-Block



# WeatherActivity.java

```
public void updateUI() {
    if (weatherReportList.size() > 0) {
        DailyWeatherReport report = weatherReportList.get(0);

        switch (report.getWeather()) {
            case DailyWeatherReport.WEATHER_TYPE_CLOUDS:
                weatherIconMini
                    .setImageDrawable(getResources()
                        .getDrawable(R.drawable.cloudy));
                weatherIcon.setImageDrawable(getResources().getDrawable(R.drawable.cloudy));
                break;
            case DailyWeatherReport.WEATHER_TYPE_RAIN:
                weatherIconMini.setImageDrawable(getResources().getDrawable(R.drawable.rainy));
                weatherIcon.setImageDrawable(getResources().getDrawable(R.drawable.rainy));
                break;
            default:
                weatherIcon.setImageDrawable(getResources().getDrawable(R.drawable.sunny));
                weatherIconMini.setImageDrawable(getResources().getDrawable(R.drawable.sunny));
        }

        weatherDate.setText(report.getFormattedDate());
        currentTemp.setText(Integer.toString(report.getCurrentTemp()));
        lowTemp.setText(Integer.toString(report.getMinTemp()));
        cityCountry.setText(report.getCityName() + ", " + report.getCountry());
        weatherDescription.setText(report.getWeather());
    }
}
```

Das switch-Statement muss noch ergänzt werden



# WeatherActivity.java

```
@Override
public void onConnected(@Nullable Bundle bundle) {
    if (ContextCompat.checkSelfPermission(
        this,
        Manifest.permission.ACCESS_FINE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(
            this,
            new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
            PERMISSION_LOCATION);
    } else {
        startLocationServices();
    }
    downloadWeatherData(LocationServices
        .FusedLocationApi
        .getLastLocation(mGoogleApiClient));
}
```

# WeatherActivity.java

```
@Override
public void onConnectionSuspended(int i) { }

@Override
public void onConnectionFailed(@NonNull ConnectionResult connectionResult) { }

@Override
public void onLocationChanged(Location location) {
    downloadWeatherData(location);
}

public void startLocationServices() {
    try {
        LocationRequest req = LocationRequest
            .create()
            .setPriority(LocationRequest.PRIORITY_LOW_POWER);
        LocationServices.FusedLocationApi
            .requestLocationUpdates(mGoogleApiClient, req, this);
    } catch (SecurityException exception) {
        Log.e(LOG_TAG, exception.getLocalizedMessage());
    }
}
```

# WeatherActivity.java

```
@Override
public void onRequestPermissionsResult(int requestCode,
                                     @NonNull String[] permissions,
                                     @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode,
                                    permissions,
                                    grantResults);

    switch (requestCode) {
        case PERMISSION_LOCATION: {
            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                startLocationServices();
            } else {
                Toast.makeText(
                    this,
                    "I can't run your location dummy - you denied permission!"
                    , Toast.LENGTH_LONG).show();
            }
        }
    }
}
```

# DailyWeatherReport.java

```
public class DailyWeatherReport {

    private static final String LOG_TAG = DailyWeatherReport.class.getSimpleName();

    public static final String WEATHER_TYPE_CLOUDS = "Clouds";
    public static final String WEATHER_TYPE_CLEAR = "Clear";
    public static final String WEATHER_TYPE_RAIN = "Rain";
    public static final String WEATHER_TYPE_WIND = "Wind";
    public static final String WEATHER_TYPE_SNOW = "Snow";

    private String cityName;
    private String country;
    private int currentTemp;
    private int maxTemp;
    private int minTemp;
    private String weather;
    private String formattedDate;

    public DailyWeatherReport(String cityName, String country, int currentTemp,
                               int maxTemp, int minTemp, String weather, String rawDate) {
        this.cityName = cityName;
        this.country = country;
        this.currentTemp = currentTemp;
        this.maxTemp = maxTemp;
        this.minTemp = minTemp;
        this.weather = weather;
        this.formattedDate = rawDateToPretty(rawDate);
    }
}
```

# DailyWeatherReport.java

```
/**
 * convert raw date into formatted date
 * @param rawDate
 */
public String rawDateToPretty(String rawDate) {

    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    SimpleDateFormat prettyFormat = new SimpleDateFormat("d. MMMM");
    Date date = null;
    try {
        date = sdf.parse(rawDate);
    } catch (ParseException e) {
        Log.e(LOG_TAG, e.getLocalizedMessage());
    }
    return prettyFormat.format(date);
}

// Getter
```

# Probelauf

AppCompat.Light

Funshine

Funshine

Today, May 1

13<sub>11</sub>

Traun, AT

Clouds

	May 1	13
	Clouds	11
	May 1	10
	Clouds	9
	May 1	10
	Clouds	10
	May 1	9
	Clear	8
	May 1	9
	Rain	9

Funshine

Today, May 1

13<sub>11</sub>

Traun, AT

Clouds

	May 1	13
	Clouds	11
	May 1	10
	Clouds	9
	May 1	10
	Clouds	10
	May 1	9
	Clear	8
	May 1	9
	Rain	9

# Nach der letzten Überarbeitung







Noch  
Fragen?